UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/616,811	07/09/2003	Samson Mkrtchyan	03226.000104.	2272

5514 7590 07/30/2009

FITZPATRICK CELLA HARPER & SCINTO
30 ROCKEFELLER PLAZA
NEW YORK, NY 10112

EXAMINER	
ZHU, RICHARD Z	

ART UNIT	PAPER NUMBER
2625	

MAIL DATE	DELIVERY MODE
07/30/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/616,811	MKRTCHYAN ET AL.	
	Examiner	Art Unit	
	RICHARD Z. ZHU	2625	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 May 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *23-28 and 86-91* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *23-28 and 86-91* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *5/22/2009*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

ETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 05/19/2009 has been entered. IDS submitted on 05/22/2009 has been entered, considered, and made of record.

Status of the Claims

2. Claims 23-28 and 86-91 are pending.

Response to Applicant's Arguments

3. **In response to "No mention is made in Reilly of notifying by the gaming machine printer the game controller coupled to the second communication port when an external device is coupled to a first communication port".**

It is unclear to the examiner as to how the applicant reach the conclusion that the reconnection mechanism facilitated by CSCP cannot possibly be used to notify a game controller by a gaming machine printer that has determined an external device is connected to

another communication port. First and foremost, contrary to applicant's assertion that asynchronous status update are send only after a host reconnects to the network printer, the section of ***Reilly*** quoted by the applicant explicitly discloses that it is the printer that facilitates reconnection to previously connected hosts and that this reconnection mechanism is used to send the status update **("IDP uses the reconnection mechanism…..to send asynchronous status update")** because the connection services layer 60 that implements the CSCP protocol, which facilitates reconnection of the printer to the host, is in itself implemented by ROM 412 of the network printer **(Col 9, Rows 10-15)**, not the host.

Second, ***Reilly*** does "remotely suggests" "notifying by the gaming machine printer the game controller coupled to the second communication port when the external device is coupled to the first communication port". ***Reilly*** discloses "CSCP facilitates reconnection to previously connected hosts. IDP **uses the reconnection mechanism** to implement remote queuing features, **to subsequently** request job data and to send asynchronous status updates to clients which support IDP" **(Col 6, Rows 45-49)**. These disclosures, in light of Figs 4-5, suggest the following: when a host wants to print a job, it will open a connection with the printer **(Fig 4, Step 5600 and Fig 5, 5701)** to make a request. In turn, the printer will detect the request and thereafter add the request to the queue **(Fig 4, 5606-5612 and Fig 5, 5703-5715, remember, bi-directional communication is conveyed through coupling via centronics ports)**. This is how the printer was able to detect **when an external device is coupled** to either Non-IDP network manager port or IDP network manager port; after all, how can a printer add something to a queue if the printer is not aware of its existence in the first place? Thereafter, the host **(when the host is a IDP host, Fig 4, 5614** and **Fig 5, 5717,**

when the host is a non-IDP host) closes said connection to wait for the printer to request transfer of print data. Similarly, the printer closes the socket that sustains the connection as a response (**See Paragraph 4 of examiner's response concerning sockets**). Thereafter, the printer dequeues the request from the print queue and issues a reconnection request (**Fig 4, 5616-5618 and Fig 5, 5719 - 5721**). Here, according to the disclosure cited at the beginning of this paragraph, reconnection request comprises the remote queuing feature (**5616 and 5719**), **to subsequently** request job data and to send asynchronous status updates **to clients which supports IDP**. Therefore, every time the printer implements reconnection request when an external device is coupled to one of the communication ports, asynchronous status update is send or otherwise notified to any and all hosts that support IDP.

Given the fact that the printer communicates with said hosts in its native IDP protocol or language, the port through which the host connects to is native and the session trusted (**See further explanation in the 6th paragraph of examiner's response**), ***Reilly*** does indeed "remotely suggests" the concept the applicant alleged it did not. All that is left is for one of ordinary skill in the art at the time of the invention is to implement a game machine with a game controller with IDP protocols as its native language for communicating with the printer such that the game machine has access to enhanced IDP features for facilitating network communication over client-server architecture because one of ordinary skill would've recognize that a network printer would better serve a plurality of networked game machines at reduced financial cost.

4. **In response to "That is, there is no linkage in Reilly between closing a socket in order to communicate with another host".**

The Berkeley Protocol **(Col 5, Rows 25-50 of *Reilly*)** is a TCP/IP protocol for bi-directional communication over a network wherein a Berkeley socket is an endpoint at the printer that allows such bidirectional communication. Usually, when a client **(For example, host computer at Fig 4, step 5600)** opens a connection in order to make a request to the printer, the printer has to create a socket or an endpoint to receive the request and thereby creating a new TCP connection by allocating resources to create said socket. When the host closes the connection **(Fig 4, 5614)**, the TCP connection is terminated and the resources allocated to the socket is released. In this manner, the socket is closed by the printer. After all, there can not be an endpoint if there is no starting point. In order to reconnect, a new connection must be open by the printer **(Col 6, Rows 40-44, CSCP being implemented by printer)** to the previously connected host making a print request.

Reilly at Column 6, Rows 45-55 in view of Figs 4-5 discloses "IDP only stores the job data request at the printer when the printer is busy printing another job and the actual job data will remain locally at the host computer. Thereafter, the network printer will call back the host computer which corresponds to the first job data request queued in the print queue after the print job is completed". Essentially, in order for the printer to reconnect to a second host, it must first complete the job it is currently processing by completing the sending and receiving of data **(Fig 4, Step 5630)** over an active connection with a first socket, allow a first host to close the connection **(Fig 4, 5632)**, terminate the connection by the printer **(A connection is terminated using close() which releases printer resource allocated to the socket or endpoint to thereby terminate the TCP connection)**, and finally issue reconnection request to open an active connection using a second socket to communicate

with the second host **(Fig 4, 5618-5620)**. See for example, the attached NPL "***Socket Programming***", Page 9 **(Steps 1-9, in particular, step 9, close server socket is exit command given by client, i.e., Step 5614 and 5632, host closes connection)**. As such, ***Reilly*** must close the connection to the first host in order to send and receive data from/to the second host.

5. **In response to "Firstly, Applicant submits that nothing in the services provided by the network printer remotely resemble reporting the communication session to the game controller when a communication session is completed as featured in the claims".**

In view of Fig 4, Steps 5614 to 5620, lets implement a practical example where a first host with a first print job is being printed and a second host with a second print job is making a request to the printer for a connection **(Assuming both host computers have IDP as its native protocol as required by Fig 4)**. Accordingly, the printer queues the print request from the second host and the second host closes the connection **(5600-5614)**. Thereafter, the printer finishes printing of the first print job and thereby finishes the transmission and reception data over a first TCP connection with the first host and closes said first TCP connection **(Fig 4, 5628-5632 in view of Col 6, Rows 45-55)**. The print request from the second printer is dequeue and a new connection employing a new socket is created to communicate with the second printer such that a second TCP connection with the second host can initiate transmission and reception of data between the printer and the second host **(Fig 4, 5618-5632)** wherein the reconnection step at 5618 comprises remote queuing features of dequeue print request from the second host, to subsequently request job data and to send asynchronous status update to **clients** which support IDP **(That is, because both first and**

second host supports IDP, a child process in the active TCP socket is created to concurrently communicate the status to both and perhaps to any other computer that supports IDP over the host, a well known feature of TCP protocol). From the stand point of the first host computer, the status update is send when the second host computer is detected by the printer to have made a request and from the stand point of the second host computer, the status update is send when the first host computer has completed its communication session. Since both computers communicate in TCP as its native language and any one of it is a trusted host, one of ordinary skill in the art at the time of the invention can fully able to implement a reporting of status update to all game controllers connected via native ports in a trusted child process or session in view of ***Stockdale*** because it would've been convenient to connect a plurality of gaming machines over a network to a network printer in the manner of ***Reilly*** in order to reduce financial burden of having a printer for a gaming machine.

Of course, this scenario is merely suggested by ***Reilly***. In all fairness to the applicant, the examiner introduces ***Shima*** to teach this explicit scenario (***Shima* also shows how status update pertaining to a queue, which is how applicant interpreted status update, is a report of communication session**).

6. **In response to "Reilly is not seen to disclose a trusted host at all, much less determining if a game controller is coupled to a communication port and establishing the communication port as a native communication port to a trusted host when the game controller is detected on the communication port**".

The examiner is reversing the previous findings that ***Reilly*** does not disclose the concept of "trusted host" and the examiner will further clarify the reasons why ***Reilly*** teaches said concept as well as concept of "native port".

According to page 15, lines 15-25 of applicant's specification: "As a native port, each communication port may communicate with games and other hosts in the game's or host's native language."

In ***Reilly,*** communication sessions between the network printer and client host computers are implemented using Imaging Device Protocol or IDP, a form of Berkeley TCP/IP. Further, the network printer distinguishes between client host computers that use IDP protocols and non-IDP protocols **(Col 3, Rows 48-65)** such that print requests made by non-IDP client host computers are also added to a print queue as if they are IDP client host computers. This is done by a non-IDP I/O network manager in cooperation with IDP emulator that emulates or translates non-IDP protocol into IDP protocol, a protocol or language that the printer can understand **(Col 5, Row 66 – Col 6, Row 24)**. Therefore, host computers that uses IDP protocol as its native language are able to communicate with the printer directly without IDP emulator **(Col 5, Rows 5-15)** because **the printer's I/O Network Manager 30 or bi-directional Centronics port is able to communicate with the host in its own native language**. As such, network manager 30 can properly be understood as a native port for handling communication with IDP host computers in host's native language.

According to Page 14, lines 1-27, of applicant's specification, "The cashless enabled game represents a trusted host for a gaming machine printer, and the communications protocol between the cashless enabled game and gaming machine printer may vary between game manufacturers. In order for the gaming machine printer to communicate with the cashless enabled games, the gaming machine printer is cognizant of multiple communication protocols required by the cashless enabled games, and **the printer is capable of recognizing a cashless enabled game coupled to the gaming machine printer through a communications port as a trusted host**.

The gaming machine printer also provides a primary second communication port and automatically disconnects the gaming machine printer from the native communication port(s) when a plug, compatible with the primary second port, is inserted into the primary second port. In addition, the gaming machine printer detects the connection to the primary second communication port, remembers that the connection was completed, and **reports the connection event to a trusted host after communications are restored to the trusted host**. The gaming machine printer only allows trusted communications to occur through the primary second port as the primary second port normally is used for downloading and or uploading information to and from the gaming machine printer without removing the gaming machine printer from the game, thus providing in-place servicing features".

To the best understanding of the examiner, the gaming machine or the host computer that implements the game controller for operating the gaming machine is inherently recognized by the game printer as the trusted host and the printer further reports connection event to a trusted host after communications are restored to the trusted host. Similarly, client

host computers that implements IDP as its native language for communicating with the printer are **also recognized as trusted host** to the network printer of ***Reilly*** because the printer is able to communication with said hosts using IDP, which allows access to enhanced IDP features that non-IDP clients or non-trusted hosts can not access **(Col 6, Rows 5-10, enhanced features are privileges of trust that non-trusted hosts are not entitle to)**, and that asynchronous status updates are send **to clients which support IDP** when communication are restored to said client host computers **(Col 6, Rows 45-50, it appears that status update is not send to non-IDP hosts. In light of Figs 4-5, CSCP reconnection mechanisms are activated when the printer reconnects to the host after its request is dequeue from the print queue, step 5618 and 5721, which is after it finishes its previous printer request and thereby closing the connection from a previous host, Col 6, Rows 49-55)**.

7. **In response to "Therefore, there is no disclosure or suggestion in Reilly of initiating by a gaming machine printer a transmission of a status to a game controller when the gaming machine printer reestablishes a connection to the game controller".**

Remember, the architecture shown in Fig 2 of ***Reilly*** is implemented by ***Reilly***'s network printer **(Col 9, Rows 10-20, ROM 412 implements the layers of architecture for interfacing the host computers)** wherein the reconnection mechanism is implemented by said architecture and therefore printer. Consider Fig 4, Step 5618-5620 and Fig 5 Step 5721 and Step 5723, the system control **(CPU 416 that controls the architecture)** issues a reconnection mechanism implemented by CSCP within connection services layer that subsequently request job data and to send asynchronous status update **(Col 6, Rows 25-55)** to

clients which support IDP **(Again, any host computer that uses IDP as its native language)**.

All that is left is for one of ordinary skill in the art at the time of the invention to implement a game machine with a game controller with IDP protocols as its native language for communicating with the printer such that it has access to enhanced IDP features for facilitating network communication over a client-server architecture, which includes receiving status update from the printer whenever the printer finishes a communication session with a first printer and starts a new communication session with a second printer.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 23-28 and 86-91 are rejected under 35 USC 103(a) as being unpatentable over ***Reilly (US 6401150 B1)*** in view of ***Stockdale et al. (US 6503147 B1)*** and ***Shima (US 7324224 B2)***.

Regarding the apparatus of Claim 23 and therefore method of Claim 24, *Reilly* discloses a network machine printer **(Fig 6, Printer 410 and see Col 9, Rows 10-20)**, comprising:

a processor **(Fig 6, CPU 416)**;

a first communication port coupled to the processor **(Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture shown in Col 5, Rows 50-65 for providing an interface for IDP ports that connects to a plurality of host computers 400 over the network, this includes bi-directional centronics port IDP network managers 30 and Parallel Port Manager 20)**;

a second communication port coupled to the processor **(Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture shown in Col 5, Rows 5-15 for providing an interface for IDP ports that connects to a plurality of host computers 400 over the network)** for connection to a host computer **(Fig 1, Host 1_0 to 1_n)**, the second communication port is a native communication port connecting the host computer as a trusted host to the network machine printer **(Col 5, Rows 10-15, these ports are able to communicate with all of the IDP features between printers and computers that uses IDP as its native protocol)**;

a memory coupled to the processor, the memory having program instructions executable by the processor stored therein **(Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture using a UNIX based program, see Col 5, Rows 15-25)**, the program instructions comprising:

determining by the game machine printer when an external device is coupled to the first communication port **(Col 8, Rows 12-25, socket service, connection service and system service cooperates with server 80 to detect incoming requests made by hosts. If a detection is made, the host, which is external, is determined to be coupled to a**

corresponding port. This process is perform by listen(), see NPL "Socket Programming" at Page 9, step 3);

notifying by the network machine printer the host computer coupled to the second communication port when the external device is coupled to the first communication port (**Col 6, Rows 45-55, sending asynchronous status updates to all host computers that support IDP as its native protocol include previously connected hosts, said status update includes number of elements within print queue 82, information indicative of various other clients who had established communication session with the network printer, Col 4, Rows 50-56**); and

disconnecting by the gaming machine printer communications from the host computer when the external device is coupled to the first communication port (**Col 5, Rows 40-41 in view of Fig 4, Step 5614. According to "Socket Programming" at page 9, a server socket or TCP connection is terminated when a client gives the command to do so. At step 5614, such command is given when the host orders the connection be closed. Col 6, Rows 40-50 in view of Fig 4, Steps 5614-5620**);

establishing by the network machine printer a communication session with the external device (**Col 6, Rows 40-45, negotiating a port connection between the network printer and a requesting client**);

reporting by the gaming machine printer the communication session to the host when the communication session is completed and communications are restored by the network machine printer to the original host computer (**Col 6, Rows 45-55. After the printing**

request is completed and hence the communication session ended, the printer reconnects with previous hosts that support IDP to provide asynchronous update reporting various parameters of previous communication session as part of its reconnection mechanism when executing a next job within the print queue, Col 4, Rows 35-56, Col 6, Rows 45-55 in view of Fig 4).

Reilly does not disclose the network printer being a gaming machine printer connected to a gaming controller.

Stockdale discloses a gaming machine printer connected to a gaming controller (**Fig 2, Printer 238 connected to the main cabinet 4 of gaming machine 2 shown in Fig 1 having a master gaming controller 200, see Fig 2 and Col 6, Rows 20-24**) connected to a processor (**Figs 2-3, Peripheral Controller 234 and Control Microprocessor 312**) having program instructions executable by the process stored within a memory (**Col 11, Rows 1-10**) for notifying the gaming controller its current peripheral configurations and status information (**Col 11, Rows 10-20 and Rows 55-65**) via an established communication port (**Fig 2, Hub 230**).

It would've been obvious to one of ordinary skill in the art at the time of the invention was made to implement the network printer of ***Reilly*** as the gaming machine printer of ***Stockdale*** connected to a gaming controller as a client host so as to execute printing operations for the plurality of peripheral devices and the gaming controller controlled machine connected to the printer over the network whereas the motivation would've been to provide an advantageous network gaming machine printer where network traffic is

advantageously reduced (***Reilly*, Col 7, Rows 64-67**) and to reduce financial cost by having a printer to serve a plurality of gaming machines and its peripherals.

The combined teachings are deficient in that status updates are only send whenever reconnection mechanism is implemented. That is, within the context of Figs 4-5, reconnection mechanism is only used whenever there are jobs remaining within the print queue. Hence, ***Reilly*** appears to suggest when the queue is empty, updates will not be available to clients that support IDP protocols.

Shima discloses a real time reporting of progressive states of a print queue or communication session to a host computer by a network printer to said host computers regardless of whether or not the queue is empty (**Figs 23-24, and see Col 3, Rows 55-65 as well as Col 14, Rows 20-36**).

Shima suggested that there is a need for user to have more control over a printer job submitted by the user (**Col 1, Row 65 – Col 2, Row 3**) to a network printer and the best way to do so would be to consistently send update information or progressively report communication sessions to the host computer (**Col 15, Rows 42-46**). One of ordinary skill in the art at the time of the invention would've recognize the advantage in sending asynchronous status update or report of communication session, which requires reconnecting to host computers support IDP via native port through trusted sessions, regardless of whether or not there is job remaining in the queue in the manner suggested by at least Figs 23-24 of ***Shima*** whereas motivation would've been to enhance network convenience for a user that is taking advantage of a network printer connected to a plurality of game machines.

Regarding the apparatus of Claim 25 and therefore method of Claim 28, ***Reilly*** discloses a network machine printer (**Fig 6, Printer 410 and see Col 9, Rows 10-20**), comprising:

a processor (**Fig 6, CPU 416**);

a communication port coupling the network machine printer to an external host (**Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture shown in Col 5, Rows 5-15 for providing an interface for a plurality of IDP ports that connects to a plurality of host computers 400 over the network**);

a memory coupled to the processor, the memory having program instructions executable by the processor stored therein (**Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture using a UNIX based program, see Col 5, Rows 15-25**), the program instructions comprising:

determining by the network machine printer the status of a communication link to the host via the communication port (**Col 8, Rows 12-25, connection service layer implemented by the printer detects an incoming service request from an IDP host or status of communication link between the host and the printer. The request is thereafter added to the print queue**); and

storing by the network machine printer the status of the network printer in a memory when the network machine printer determines that the communication link is interrupted (**Col 6, Rows 40-55 and Col 4, Rows 45-60 in view of Col 8, Rows 12-25. When an incoming host made a print request, the request is accepted by being saved to the print queue so**

that the printer remembers which host made the request. Thereafter, the host closes the connection or otherwise disrupts the communication link);

transmitting by the network machine printer the status of the network machine printer to the host when the communication link is reestablished by the network machine printer (**Col 6, Rows 52-55, call back the host computer that requested the first print job. Col 4, Rows 50-56, said call back process includes current status of the printer pertaining to various updated parameters**);

Reilly does not disclose the network printer being a gaming machine printer connected to a game controller and locking the status of the gaming machine printer in the non-volatile memory when the gaming machine printer determines that the communication link is interrupted.

Stockdale discloses a gaming machine printer connected to a gaming controller (**Fig 2, Printer 238 connected to the main cabinet 4 of gaming machine 2 shown in Fig 1 having a master gaming controller 200, see Fig 2 and Col 6, Rows 20-24**) connected to a processor (**Figs 2-3, Peripheral Controller 234 and Control Microprocessor 312**) having program instructions executable by the process stored within a memory (**Col 11, Rows 1-10**) for notifying the gaming controller its current peripheral configurations and status information (**Col 11, Rows 10-20 and Rows 55-65**) via an established communication port (**Fig 2, Hub 230**).

Stockdale further discloses locking in a memory status information of a gaming machine printer when it is determined that a communication link with a peripheral is

interrupted (**Col 15, Row 60 – Col 16, Row 5, interrupting communication link to a first host or peripheral to connect to a second host or peripheral. Col 16, Rows 40-55, logging error status of the printer peripheral when communication link with a peripheral is interrupted naturally requiring saving said log into some sort of memory**).

Stockdale suggested that storing in a non-volatile memory status information of the gaming machine printer ensures that in the event of some critical malfunction (**Col 17, Row 60 – Col 18, Row 8**) such as loss of communication link with a functioning peripheral (**Col 17, Rows 1-20**) ensures critical status information is not lost.

One of ordinary skill in the art, facing with the problem that either a critical event such as communication link interruption due to component or peripheral failures or non-critical event such as communication link interruption due to standard operation to connect to another peripheral or host could occur at anytime, would look to ***Stockdale*** for solution since it suggested storing status information of a printer in non-volatile memory would ensure such information be preserved for future use; for example, when reconnecting to a previously connected host and said host requests status update of the printer as taught by ***Reilly***.

It would've been obvious to one of ordinary skill in the art at the time of the invention was made to implement the network printer of ***Reilly*** as the gaming machine printer of ***Stockdale*** connected to a gaming controller as a client host so as to execute printing operations for the plurality of peripheral devices and the gaming controller to save in non-volatile memory status information of the gaming machine printer when a malfunction such as interrupted communication link occurs whereas the motivation would've been to provide

an advantageous network gaming machine printer where network traffic is advantageously reduced (***Reilly*, Col 7, Rows 64-67**) and in event of malfunctions during a certain on going operation, critical status information can be used to determine the state or status of the gaming machine before the interruption (***Stockdale*, Col 18, Rows 5-8**).

Regarding the apparatus of Claim 27 and therefore method of Claim 26, *Reilly* discloses a network machine printer (**Fig 6, Printer 410 and see Col 9, Rows 10-20**), comprising:

a processor (**Fig 6, CPU 416**);

a plurality of communication ports coupled to the processor (**Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture shown in Col 5, Rows 5-15 for providing an interface for IDP ports that connects to a plurality of host computers 400 over the network**);

a memory coupled to the processor, the memory having program instructions executable by the processor stored therein, (**Fig 6, ROM 412 and see Col 9, Rows 12-15, ROM being used to implement network layer architecture using a UNIX based program, see Col 5, Rows 15-25**) the program instructions comprising:

for each of the plurality of communication ports, determining by the network machine printer if a host is coupled to the communication ports (**Col 6, Rows 13-24, IDP emulator for detecting incoming communication from a newly connected client, see also Col 5, Rows 32-42 + Col 8, Rows 12-25, connection service layer implemented by the printer**

detects an incoming service request from an IDP host or status of communication link between the host and the printer. The request is thereafter added to the print queue);

establishing by the network machine printer the communication port as a native communication port (**Col 5, Rows 5-15, IDP I/O Network Manager 30 is considered a native port for connecting to IDP clients because it communicates with clients in its native IDP protocols**) that is disconnected from the host prior to performing a separate processing request by the network machine printer (**Fig 4, Step 5632-> 5614->5620 in view of Col 6, Rows 45-55, in order to process a second print request, the printer must complete a first print request. Terminate socket to a first host that made the first request when the host closes the connection at 5632. Create a new socket to the second host via reconnection mechanism, which comprises sending asynchronous status update to all previously connected host computers that supports IDP by creating a plurality child process that enable concurrent connections to said plurality of computers**), the native port for connection to the host as a trusted host when the host is detected on the communication port (**Col 5, Rows 10-15, these ports are able to communicate with all of the IDP features between printers and computers that uses IDP as its native protocol, in contrast to non-IDP clients that does not have access to enhanced IDP features**).

Reilly does not disclose the network printer being a gaming machine printer connected to a gaming controller as a trusted host to the gaming machine printer.

Stockdale discloses a gaming machine printer connected to a gaming controller (**Fig 2, Printer 238 connected to the main cabinet 4 of gaming machine 2 shown in Fig 1**

having a master gaming controller 200, see Fig 2 and Col 6, Rows 20-24) connected to a processor (**Figs 2-3, Peripheral Controller 234 and Control Microprocessor 312**) having program instructions executable by the process stored within a memory (**Col 11, Rows 1-10**) for notifying the gaming controller its current peripheral configurations and status information (**Col 11, Rows 10-20 and Rows 55-65**) via an established communication port (**Fig 2, Hub 230**).

It would've been obvious to one of ordinary skill in the art at the time of the invention was made to implement the network printer of ***Reilly*** as the gaming machine printer of ***Stockdale*** connected to a gaming controller as a client host so as to execute printing operations for the plurality of peripheral devices and the gaming controller controlled machine connected to the printer over the network whereas the motivation would've been to provide an advantageous network gaming machine printer where network traffic is advantageously reduced (***Reilly,*** **Col 7, Rows 64-67**).

Regarding Claims 86-91, ***Reilly*** discloses the plurality of communication port or first and second communication port are communication ports selected from the group including a serial port, a parallel port, a Universal Serial Bus (USB) port and an Ethernet port (**Col 5, Rows 5-15, Rows 50-65, and Col 5, Row 66 – Col 6, Row 12, parallel port that supports Ethernet protocols**).

Conclusion

10. US 5564110 A and US 6560660 B1 discloses the plugging of a peripheral to either a printer or a computer, the peripheral is detected, communication protocols between the peripheral and the host (printer or computer) is verified, and connection is declined or accepted.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to examiner Richard Z. Zhu whose telephone number is 571-270-1587 or examiner's supervisor King Y. Poon whose telephone number is 571-272-7440. Examiner Richard Zhu can normally be reached on Monday through Thursday, 0630 - 1700.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

RZ^2
06/16/2009

Richard Z. Zhu
Assistant Examiner
Art Unit 2625

/King Y. Poon/

Supervisory Patent Examiner, Art Unit 2625